Chapter 2
Ins and Outs of Network-Oriented Modeling

Abstract Network-Oriented Modeling has successfully been applied to obtain network models for a wide range of phenomena, including Biological Networks, Mental Networks, and Social Networks. In this chapter, it is discussed how the interpretation of a network as a causal network and taking into account dynamics in the form of temporal-causal networks, brings more depth. Thus main characteristics for a network structure are obtained: Connectivity in terms of the connections and their weights, Aggregation of multiple incoming connections in terms of combination functions, and Timing in terms of speed factors. The basics and the scope of applicability of such a Network-Oriented Modelling approach are discussed and illustrated. This covers, for example, Social Network models for social contagion or information diffusion, and Mental Network models for cognitive and affective processes. From the more fundamental side, it will be discussed how emerging network behavior can be related to network structure.

Keywords Network-Oriented Modeling · Temporal-causal network

2.1 Introduction

Network-Oriented Modeling is a relatively new way of modeling that is especially useful to model intensively interconnected and interactive processes. It has been applied to model networks for biological, mental, and social processes, and still more. The aim of this chapter is to discuss the ins and outs of this modeling perspective in more detail, without considering network reification yet, as that will be the subject of Chap. 3. It is discussed how the interpretation of a network as a causal network and taking into account dynamics brings more depth in the Network-Oriented Modeling perspective, leading to the notion of temporal-causal network as introduced in (Treur 2016). In a temporal-causal network, nodes represent states with values that vary over time, and connections represent causal relations describing how states affect each other.

J. Treur, *Network-Oriented Modeling for Adaptive Networks: Designing Higher-Order Adaptive Biological, Mental and Social Network Models*, Studies in Systems, Decision and Control 251, https://doi.org/10.1007/978-3-030-31445-3_2

The wide scope of applicability (Treur 2016, 2017) of such a Network-Oriented Modelling approach will be discussed and illustrated. This covers, for example, network models for principles of social contagion or information diffusion, and network models for mental processes. When network reification as introduced in more detail in Chap. 3 is also taken into account, many kinds of adaptive network models are covered, for example for principles of evolving social networks, such as the homophily principle, or for Hebbian learning in Mental Networks. From the methodological side, it will be discussed how mathematical analysis can be used to identify the relation between emerging behaviour of the network and network structure.

In this chapter, in Sect. 2.2 first the conceptual background of Network-Oriented Modeling is discussed, leading to a conceptual representation of a temporal-causal network, which defines such a network. Next, in Sect. 2.3 the numerical foundation is discussed, including a precise definition of a numerical representation by which a temporal-causal network model gets its intended dynamic semantics, and which can be used for simulation and analysis. Section 2.4 introduces role matrices as a useful specification format for temporal-causal networks. In Sect. 2.5 the interesting challenge to determine how emerging network behaviour relates to network structure and some results on this relation are briefly discussed. In Sect. 2.6 the scope of applicability is discussed. Finally, Sect. 2.7 is a discussion.

2.2 Network-Oriented Modeling: Conceptual Background

Network-Oriented Modeling is applied in a wide variety of areas. The general pattern is that some type of process in some domain *X* is described by a network structure, and this type of network is called an *X* Network or *X* Network model. Note that such a network is considered as a modelling concept, not as reality. Some examples are:

- Modeling the dynamics of propagation of chemical activity in cells based on the concentration levels of chemicals by Biological Network models
- Modeling the dynamics of propagation of neural activity based on activation levels of neurons by Neural Network models
- Modeling the dynamics of propagation of mental activity based on engaging mental states by Mental Network models
- Modeling the dynamics of propagation of individual activity based on activation of personal states by Social Network models; e.g.,
  - Information diffusion; e.g., in social media
  - Opinion spread; e.g., in political campaigns
  - Emotion contagion; e.g., one smile triggering the other
  - Activity contagion; e.g., following each other

These are just four types of domains *X* where processes, in reality, are modelled by network models, which then can be called *X* Networks with *X* = Biological, Neural, Mental, or Social.

2.2.1 The Unifying Potential of Networks

As an illustration, consider the following two examples, one for a Biological Network, and one for a Mental Network. The example of a Biological Network shown in Fig. 2.1 describes how bacteria generate and regulate their behaviour on the one hand based on their genetical background as encoded in their DNA, and on the other hand based on the situational context of the environment; see also Jonker et al. (2008). For the general perspective on modelling the cell's metabolic and life processes as biochemical networks ('the dynamic biochemical networks of life'), see also Westerhoff et al. (2014a, b). For example:

> Living organisms persist by virtue of complex interactions among many components organized into dynamic, environment-responsive networks that span multiple scales and dimensions. Biological networks constitute a type of information and communication technology (ICT): they receive information from the outside and inside of cells, integrate and interpret this information, and then activate a response. Biological networks enable molecules within cells, and even cells themselves, to communicate with each other and their environment. (Westerhoff et al. 2014b, p. 1)

As a second example, the Mental Network shown in Fig. 2.2 describes how human behaviour is generated and regulated by desires and intentions, and beliefs

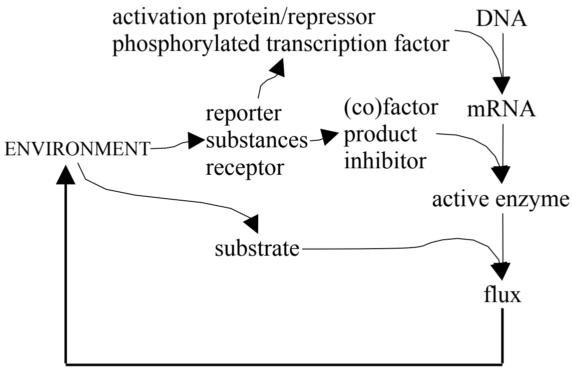

Fig. 2.1 Example of a Biological Network for bacterial behaviour based on its biochemistry; adapted picture from Jonker et al. (2008), Fig. 1 left hand side, p. 3

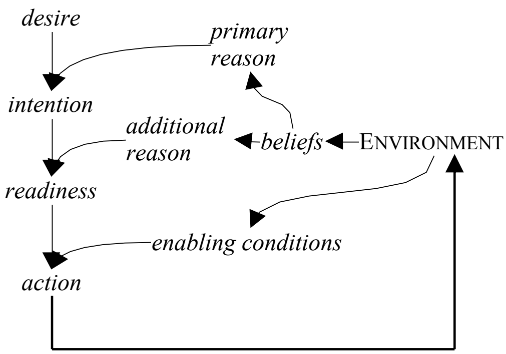

Fig. 2.2 Example of a Mental Network for behaviour based on Beliefs, Desires and Intentions (BDI); adapted picture from Jonker et al. (2008), Fig. 1 right hand side, p. 3

about the environment. Within Philosophy of Mind, Kim (1996) describes Mental Networks based on causal relations as follows:

> Mental events are conceived as nodes in a complex causal network that engages in causal transactions with the outside world by receiving sensory inputs and emitting behavioral outputs. (Kim 1996, p. 104)

As can be noted similar network structures may describe different types of processes; see the isomorphic structures in Figs. 2.1 and 2.2, where actually the latter is a mirror image of the former. The Network-Oriented perspective provides a form of unification so that different types of processes become comparable, and we can, for example, compare the processes underlying human intelligence and behaviour to the processes underlying bacterial behaviour, as described in more detail in Jonker et al. (2002, 2008), Westerhoff et al. (2014b). For example:

> We have become accustomed to associating brain activity – particularly activity of the human brain – with a phenomenon we call "intelligence." Yet, four billion years of evolution could have selected networks with topologies and dynamics that confer traits analogous to this intelligence, even though they were outside the intercellular networks of the brain. Here, we explore how macromolecular networks in microbes confer intelligent characteristics, such as memory, anticipation, adaptation and reflection and we review current understanding of how network organization reflects the type of intelligence required for the environments in which they were selected. We propose that, if we were to leave terms such as "human" and "brain" out of the defining features of "intelligence," all forms of life – from microbes to humans – exhibit some or all characteristics consistent with "intelligence". (Westerhoff et al. 2014b, p. 1)

The emphasis in this quote is on how not only in the brain, but even in the smallest life forms network structure, organisation, and dynamics are used to realise many if not all aspects of intelligence.

This unifying perspective of networks for different domains can be seen in many cases. For example, a politician such as Boris Johnson can be seen as a big influencer for the population in the UK, for example, concerning the Brexit dilemma. Such an influencing process can be described by social contagion in a Social Network; in the same way a flock of sheep following a leader sheep can be described by social contagion in a similar network, where the leading sheep is the big influencer.

As network structures for different domains may look similar, this suggests that there is a high potential for unification and exchange across different domains. For example, can we learn more about Mental Networks by studying Social Networks? Or can we develop Network Theory from a unified perspective that can be applied in both areas, or even in more areas? These questions indicate some of the promises and challenges in what nowadays is called Network Science.

2.2.2 On the Meaning of the Basic Elements in a Network

There are, however, some issues that may have to be addressed to enable the further development of this perspective of a unified Network Science. A main issue is that not every network may have the same form concerning definition and semantics. Then unification may be not so easy. What actually is a network? What does a node mean? How should we interpret what a connection is or does? Are all connections considered equal? And what if there are multiple connections to one node? Should we interpret this as a kind of conjunction (AND), or disjunction (OR), or maybe something in between, like some average; then what kind of average?

Is a network just an abstract graph structure with nodes and connections and nothing more, and in particular no further semantics? Then in fact Network Science = Graph Theory, which is an already existing area within Mathematics, and Network-Oriented Modeling could also be called Graph-Oriented Modeling. This perspective may provide a relevant stream, but will not be sufficient to further develop Network Science. For many applications, just a graph structure with only nodes and connections seems seriously underspecifying what is intended.

In many examples of applications of networks, such as those mentioned above, a notion of dynamics plays an important role. Shouldn't such dynamics be part of the definition or semantics of a network? These dynamics can concern dynamics of states (dynamics *within* a network: for example, diffusion or contagion of opinions or emotions in a network), but also dynamics of the network structure itself (dynamics *of* a network: for example, adaptive or evolving networks describing changing relationships between persons). Dynamics has a direct relation to causal relations describing how one state affects the other. The notions of dynamics and causality are fundamental for practically all scientific disciplines; these notions play an important unifying role in science and can be found in most of the scientific literature. Causal relations vary from how hitting a ball causes movement of the ball to how certain beliefs cause certain behaviour or how joining forces in a social movement causes a change in society, to name just a few cases.

2.2.3 Meaning as Defined by the Notion of Temporal-Causal Network

For the perspective on Network Science addressed in the current chapter these notions of causality and dynamics have been incorporated and are part of a more refined structure and semantics of the considered networks. More specifically, the nodes in a network are interpreted here as states (or state variables) that vary over time, and the connections are interpreted as causal relations that define how each state can affect other states over time. To acknowledge this perspective of dynamics and causality on networks, this type of network has been called a *temporal-causal network* (Treur 2016). Many examples of applications have demonstrated that all types of domains as listed above can be covered in this way; e.g., Treur (2016). In Sect. 2.6 below this wide applicability is briefly discussed; see also Treur (2017).

So, is there still some relevant graph perspective? A conceptual representation of a temporal-causal network model by a *labeled* graph still provides a fundamental basis. More specifically, a conceptual representation of a temporal-causal network model in the first place still involves representing in a declarative manner states and connections between them that, as discussed earlier, represent (causal) impacts of states on each other, as assumed to hold for the application domain addressed. This part of a conceptual representation is often depicted in a *conceptual picture* by a graph with nodes and directed connections. However, a *full conceptual representation* of a temporal-causal network model also includes a number of labels for such a graph. First, in reality, not all causal relations are equally strong, so some notion of strength of a connection is used as a label for connections. Second, when more than one causal relation affects a state, some way to aggregate multiple causal impacts on a state is used as a label for states. Third, a notion 'speed of change' is used for timing of the processes for a state. These three notions define the characteristics of the network structure; they are summarized as

(a) **Connectivity**

- connection weights from a state X to a state Y, denoted by $\boldsymbol{\omega}_{X,Y}$

(b) **Aggregation**

- a combination function for each state Y, denoted by $\mathbf{c}_Y(..)$

(c) **Timing**

- a speed factor for each state Y, denoted by $\boldsymbol{\eta}_Y$

They make the graph of states and connections a labeled graph (see Fig. 2.3), form the defining structure of a temporal-causal network model in the form of a

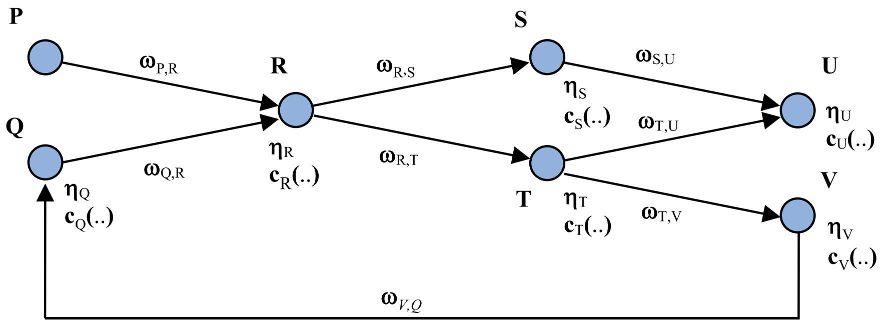

Fig. 2.3 Conceptual representation of a simple example temporal-causal network as a labeled graph, with states **P** to **V** and for each connection from X to Y *connectivity labels* (a) in terms of connection weights $\boldsymbol{\omega}_{X,Y,}$ and for each state Y *aggregation labels* (b) in terms of combination functions $\mathbf{c}_Y(..)$, and *timing labels* (c) in terms of speed factors $\boldsymbol{\eta}_Y$

Table 2.1 Conceptual representation of a temporal-causal network model

Concepts	Notation	Explanation
States and connections	X, Y, $X \rightarrow Y$	Describes the nodes and links of a network structure (e.g., in graphical or matrix format)
Connection weight	$\boldsymbol{\omega}_{X,Y}$	The *connection weight* $\boldsymbol{\omega}_{X,Y} \in [-1, 1]$ represents the strength of the causal impact of state X on state Y through connection $X \rightarrow Y$
Aggregating multiple impacts on a state	$\mathbf{c}_Y(..)$	For each state Y (a reference to) a *combination function* $\mathbf{c}_Y(..)$ is chosen to combine the causal impacts of other states on state Y
Timing of the effect of causal impact	$\boldsymbol{\eta}_Y$	For each state Y a *speed factor* $\boldsymbol{\eta}_Y \geq 0$ is used to represent how fast a state is changing upon causal impact

conceptual representation; see also Table 2.1. Note that also connections from a state to itself are allowed, although often they are not depicted in a conceptual representation as shown in Fig. 2.3. Such connections can be used to give the state a more persistent character, as the old values are reused all the time. This may be relevant, in particular, for learning or adaptation.

Combination functions, in general, are similar to the functions used in a static manner in the (deterministic) Structural Causal Model perspective described, for example, in Mooij et al. (2013), Pearl (2000), Wright (1921), but in the Network-Oriented Modelling approach described here they are used in a dynamic manner. For example, Pearl (2000), p. 203, denotes nodes by V_i and combination functions by f_i; he also points at the issue of underspecification for aggregation of multiple connections mentioned in Sect. 2.2 above, as in the often used graph representations the role of combination functions f_i for nodes V_i, is lacking:

> Every causal model *M* can be associated with a directed graph, *G(M)* (…) This graph merely identifies the endogeneous and background variables that have a direct influence on each V_i; it does not specify the functional form of f_i. (Pearl 2000, p. 203)

Therefore, if a graph representation is used, at least aggregation in terms of combination functions should be incorporated as labels, as indeed is done for temporal-causal networks, in order to avoid this problem of underspecification. That is the reason why aggregation in terms of combination functions is part of the definition of the network structure for temporal-causal networks, in addition to connectivity in terms of connections and their weights and timing in terms of speed factors.

Combination functions can have different forms, as there are many different approaches possible to address the issue of aggregating multiple impacts. For this aggregation, a library is available with a number of standard combination functions as options, but also own-defined functions can be added.

2.2.4 Biological, Mental and Social Domains Ask for Networks

In Sect. 2.1 it already was discussed how 'the dynamic biochemical networks of life' (Westerhoff et al. 2014a) are fundamental to describe life forms in the biological domain. For the mental domain, the mechanisms found within the area of Cognitive and Social Neuroscience also show how many parts in the brain have connections which are adaptive and often form cyclic pathways; such cycles are assumed to play an important role in many mental processes; see also Bell (1999), Potter (2007). It has been pointed out that to address such cyclic effects, a dynamic and adaptive perspective on causality is needed; e.g., Scherer (2009). Also, by Kim (1996) it is claimed that Mental Networks display cyclic network structures:

> (…) to explain what a given mental state is, we need to refer to other mental states, and explaining these can only be expected to require reference to further mental states, on so on – a process that can go on in an unending regress, or loop back in a circle. (Kim 1996, pp. 104–105)

For the social domain, intense interaction between persons also takes place based on mutual and usually cyclic relationships, by which they affect each other. Just one example from the context of modelling social systems or societies can be found in Naudé et al. (2008), where it is claimed that '*relational, network-oriented modelling approaches* are needed' to address human social complexity.

So, from the areas of biological processes (Westerhoff et al. 2014a, b), mental processes (Bell 1999; Kim 1996; Potter 2007; Scherer 2009) and social processes (Naudé et al. 2008), a notion of network is suggested as a basis of modeling, where connections between states or persons describe how they affect each other, thereby strongly suggesting causality and dynamics as crucial notions.

2.3 Numerical Representation of a Temporal-Causal Network

In this section, the numerical-mathematical foundations of temporal-causal networks are discussed in more detail. In Sect. 2.2 the choice made on how networks are interpreted conceptually was discussed based on the notions of temporality and causality, thus indicating semantics for networks based on the notion of temporal-causal network.

2.3.1 Numerical-Mathematical Formalisation

In the current section the interpretation based on temporality and causality is expressed in a formal-numerical way, thus associating semantics to any conceptual temporal-causal network specification in a detailed numerical-mathematically defined manner. This is done by showing how a conceptual representation as discussed in Sect. 2.2, based on states and connections enriched with labels for (a) connectivity (by connection weights), (b) aggregation (by combination functions), and (c) timing (by speed factors), defines a numerical representation (Treur 2016, Chap. 2). This is shown in Table 2.2, where Y is any state in the network and $X_1, \ldots, X_k$ are the states with outgoing connections to Y.

The difference equations in the last row in Table 2.2 form the numerical representation of a temporal-causal network model and can be used for simulation and mathematical analysis. They can also be written in differential equation format and are called the *basic difference or differential equations*:

$$\begin{aligned} Y(t+\Delta t) &= Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)]\Delta t \\ \mathbf{d}Y(t)/\mathbf{d}t &= \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)] \end{aligned} \tag{2.1}$$

Table 2.2 Numerical representation of a temporal-causal network model

concept	Representation	Explanation
State values over time t	$Y(t)$	At each time point t each state Y in the model has a real number value in [0, 1]
Single causal impact	$\mathbf{impact}_{X,Y}(t) = \boldsymbol{\omega}_{X,Y}\, X(t)$	At t state X with connection to state Y has an impact on Y, using connection weight $\boldsymbol{\omega}_{X,Y}$
Aggregating multiple impacts	$\mathbf{aggimpact}_Y(t)$ $= \mathbf{c}_Y(\mathbf{impact}_{X_1,Y}(t), \ldots, \mathbf{impact}_{X_k,Y}(t))$ $= \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t))$	The aggregated causal impact of multiple states X_i on Y at t, is determined using a combination function $\mathbf{c}_Y(V_1, \ldots, V_k)$ and apply it to the k single causal impacts
Timing of the causal effect	$Y(t+\Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) - Y(t)]\Delta t$ $=$ $Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)]\Delta t$	The causal impact on Y is exerted over time gradually, using speed factor $\boldsymbol{\eta}_Y$; here the X_i are all states from which state Y has incoming connections

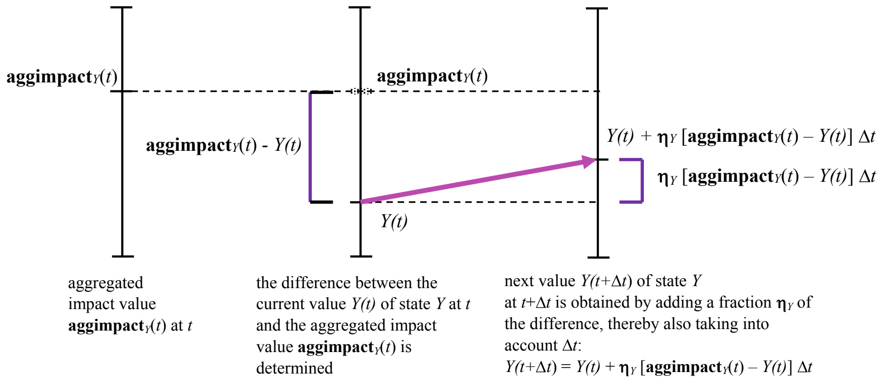

Fig. 2.4 How **aggimpact**$_Y$(t) makes a difference for state $Y(t)$ in the time step from t to $t + \Delta t$, using speed factor $\boldsymbol{\eta}_Y$ and taking into account step size Δt

This can be considered an interpretation of a network based on causality and dynamics as expressed in a formal-numerical way, thus associating semantics to any conceptual temporal-causal network representation in a detailed numerical-mathematically defined manner. Table 2.2 shows how a conceptual representation based on states and connections enriched with labels for connection weights, combination functions, and speed factors, can be transformed into a numerical representation (Treur 2016, Chap. 2). A more detailed explanation of this difference equation format, taken from Treur (2016), Chap. 2, pp. 60–61, is as follows; see also Fig. 2.4. The aggregated impact value **aggimpact**$_Y$**(**t**)** at time t pushes the value of Y up or down, depending on how it compares to the current value of Y. So, **aggimpact**$_Y$**(**t**)** is compared to the current value $Y(t)$ of Y at t by taking the difference between them (also see Fig. 2.4):

$$\mathbf{aggimpact}_Y(t) - Y(t)$$

If this difference is positive, which means that **aggimpact**$_Y$**(**t**)** at time t is higher than the current value of Y at t, in the time step from t to $t + \Delta t$ (for some small Δt) the value $Y(t)$ will increase in the direction of the higher value **aggimpact**$_Y$**(**t**)**. This increase is done proportional to the difference, with proportion factor $\boldsymbol{\eta}_Y$ Δt: the increase is (see Fig. 2.4):

$$\boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) - Y(t)]\Delta t$$

By this format, the network structure characteristic $\boldsymbol{\eta}_Y$ indeed acts as a speed factor by which it can be specified how fast state Y should change upon causal impact.

2.3.2 Combination Functions as Building Block for Aggregation

Often used examples of combination functions are the ones listed below: the *identity* **id**(.) for states with impact from only one other state, the *scaled maximum and minimum* $\mathbf{smax}_\lambda$(.) and $\mathbf{smin}_\lambda$(.), the *scaled sum* $\mathbf{ssum}_\lambda$(.) with scaling factor λ, the *advanced logistic sum* combination function $\mathbf{alogistic}_{\sigma,\tau}$(..) with steepness σ and threshold τ, and the Euclidean combination function $\mathbf{eucl}_{n,\lambda}$(.) where n is the order (which can be any nonzero natural number, but also any positive real number), and with scaling factor λ:

- *the identity* function for states with impact from only one other state

$$\mathbf{id}(V) = V$$

- the *scaled maximum and minimum* with scaling factor λ

$$\mathbf{smax}_\lambda(V_1, \ldots, V_k) = \mathbf{max}(V_1, \ldots, V_k)/\lambda$$
$$\mathbf{smin}_\lambda(V_1, \ldots, V_k) = \mathbf{min}(V_1, \ldots, V_k)/\lambda$$

- *scaled sum* with scaling factor λ

$$\mathbf{ssum}_\lambda(V_1, \ldots, V_k) = \frac{V_1 + \cdots + V_k}{\lambda}$$

- the *advanced logistic sum* combination function with steepness σ and threshold τ

$$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = \left[\frac{1}{1 + \mathrm{e}^{-\sigma(V_1 + \ldots + V_k - \tau)}} - \frac{1}{1 + \mathrm{e}^{\sigma\tau})}\right](1 + \mathrm{e}^{-\sigma\tau})$$

- the Euclidean combination function $\mathbf{eucl}_{n,\lambda}$(.) where n is the order (which can be any nonzero natural number, but also any positive real number), and with scaling factor λ:

$$\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) = \sqrt[n]{\frac{V_1^n + \cdots + V_k^n}{\lambda}}$$

Scaling factors λ are used to normalise the values so that they fit in the intended interval for their values (usually the [0, 1] interval).

Note that for $\lambda = 1$, the scaled sum function is just the sum function **sum(..)**, and this sum function can also be used as identity function in case of just one incoming connection. Furthermore, note that for $n = 1$ (first-order Euclidean combination function) we get the scaled sum function:

$$\mathbf{eucl}_{1,\lambda}(V_1, \ldots, V_k) = \mathbf{ssum}_{\lambda}(V_1, \ldots, V_k)$$

For $\boldsymbol{n} = 2$ it is the second-order Euclidean combination function $\mathbf{eucl}_{2,\lambda}$**(..)** defined by:

$$\mathbf{eucl}_{2,\lambda}(V_1, \ldots, V_k) = \sqrt{\frac{V_1^2 + \ldots + V_k^2}{\boldsymbol{\lambda}}}$$

This second-order Euclidean combination function is also often applied in aggregating the error value in optimisation and in parameter tuning using the root-mean-square deviation (RMSD), based on the Sum of Squared Residuals (SSR).

Combination functions as shown above are called *basic combination functions*. There is a *combination function library* containing these basic combination functions. Up till now the library contains 35 basic combination functions. However, it can easily be extended if the designer needs another combination function. For any network model some number *m* of them can be selected (usually just one or two, or at most a handful); they are represented in a standard format as $\mathrm{bcf}_1(..)$, $\mathrm{bcf}_2(..)$, …, $\mathrm{bcf}_m(..)$. In principle, they use parameters $\boldsymbol{\pi}_{1,i,Y}, \boldsymbol{\pi}_{2,i,Y}$ such as the $\boldsymbol{\lambda}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\tau}$ in the examples above. Including these parameters, the standard format used for basic combination functions is (with V_1, …, V_k the single causal impacts):

$$\mathrm{bcf}_i\left(\boldsymbol{\pi}_{1,i,Y}, \boldsymbol{\pi}_{2,i,Y}, V_1, \ldots, V_k\right)$$

For each state *Y* just one basic combination function can be selected, but also a weighted average of them can be selected according to the following format

$$\begin{aligned} &\mathbf{c}_Y(\boldsymbol{\pi}_{1,1,Y}, \boldsymbol{\pi}_{2,1,Y}, \ldots, \boldsymbol{\pi}_{1,m,Y}, \boldsymbol{\pi}_{2,m,Y}, \ldots, V_1, \ldots, V_k) \\ &= \frac{\boldsymbol{\gamma}_{1,Y}\mathrm{bcf}_1\left(\boldsymbol{\pi}_{1,1,Y}, \boldsymbol{\pi}_{2,1,Y}, V_1, \ldots, V_k\right) + \cdots + \boldsymbol{\gamma}_{m,Y}\mathrm{bcf}_m\left(\boldsymbol{\pi}_{1,m,Y}, \boldsymbol{\pi}_{2,m,Y}, V_1, \ldots, V_k\right)}{\boldsymbol{\gamma}_{1,Y} + \cdots + \boldsymbol{\gamma}_{m,Y}} \end{aligned} \tag{2.2}$$

with *combination function weights* $\boldsymbol{\gamma}_{i,Y}$. Selecting only one of them for state *Y*, for example, $\mathrm{bcf}_i(..)$, is done by putting weight $\boldsymbol{\gamma}_{i,Y} = 1$ and the other weights 0. This is a convenient way to indicate combination functions for a specific network model. The function $\mathbf{c}_Y$**(..)** can just be indicated by the weight factors $\boldsymbol{\gamma}_{i,Y}$ and the parameters $\boldsymbol{\pi}_{i,j,Y}$. Note that in (2.2) the different basic combination functions are assumed to share the same variables V_1, …, V_k. If that is not intended, some functions may have to be adapted by adding auxiliary variables to get this right. An example of this can be found in Chap. 5.

So, the concepts $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\eta}_Y$, $\boldsymbol{\gamma}_{i,Y}$, $\boldsymbol{\pi}_{i,j,Y}$ (all denoted by bold small Greek letters) represent the different characteristics of a network's structure. Together they fully define the network structure. They are summarised in Table 2.3. In Sect. 2.4 it is

Table 2.3 The characteristics defining the structure of a temporal-causal network model

Concept	Notation	Explanation
Connection weight	$\boldsymbol{\omega}_{X,Y}$	Specifies the strength of the connection from state X to state Y
Speed factor	$\boldsymbol{\eta}_Y$	Describes speed of change of state Y upon received causal impact
Combination function weight	$\gamma_{i,Y}$	Determines which combination function(s) are used for state Y
Combination function parameter	$\pi_{i,j,Y}$	The value of the jth parameter of the ith combination function for Y

Table 2.4 Connection matrix of the example of Fig. 2.3

Connection matrix		X_1	X_2	X_3	X_4	X_5	X_6	X_7
		P	Q	R	S	T	U	V
X_1	P			0.8				
X_2	Q			1				
X_3	R				0.9	1		
X_4	S						1	
X_5	T						0.7	1
X_6	U							
X_7	V		1					

shown how the format of role matrices can be used to specify these characteristics for a network model's structure.

For proper functioning of Euclidean combination functions, some constraints are used. First, in general, this function is only applied when all connection weights are positive, except in the specific case that $\boldsymbol{n}$ is an odd natural number. Moreover, also a constraint on the scaling factor $\boldsymbol{\lambda}$ is used. When no weights are negative, the maximal value of the outcome is achieved when for each X_i it holds $X_i(t) = 1$; then the maximal outcome is $\left((\Sigma_i \boldsymbol{\omega}^{\mathbf{n}}_{X_i,Y})/\boldsymbol{\lambda}\right)^{1/\boldsymbol{n}}$. To keep the outcomes within the [0, 1] interval 1, the scaling factor $\boldsymbol{\lambda}$ should be equal to or at least the sum of the $\boldsymbol{n}$th powers of all weights: $\boldsymbol{\lambda} \geq \Sigma_i \boldsymbol{\omega}^{\boldsymbol{n}}_{X_i,Y}$. In such cases the standard value $\boldsymbol{\lambda} \geq \Sigma_i \boldsymbol{\omega}^{\boldsymbol{n}}_{X_i,Y}$ is often used as a form of *normalisation*. All this also applies to scaled sum functions, as this is the case $\boldsymbol{n} = 1$.

2.4 Role Matrices to Specify a Network Model

To specify a network model being designed, often matrices are a useful means. The first type of matrix sometimes used is a *connection matrix*. This is a square matrix with on each of the two dimensions all states of the network, say $X_1, \ldots, X_n$. In cell (i, j) of the matrix in row i and column j (also denoted by $\mathbf{m}(i, j)$ with $\mathbf{m}$ the name of the matrix) it is indicated whether or not there is a connection from state X_i to state X_j (1 or 0) or the value of the weight of this connection is $(\boldsymbol{\omega}_{X_i,X_j})$. To get the idea, first the example shown in Fig. 2.3 is considered; see Table 2.4 for the connection matrix. For example, the 0.9 in cell (3, 4) indicates that there is a connection from X_3 to X_4 with weight 0.9.

Next, an example of a Social Network addressing social contagion (e.g., of opinions or emotions) is used as illustration: the (fully connected) network shown Fig. 2.5 with connection weights as shown in the square matrix Table 2.5. For example, in cell (4, 2) it is indicated that there is a connection from X_4 to X_2 with weight 0.15.

However, as described by the connection matrix above, the connection weights form only one of the network characteristics to specify a temporal-causal network model. The other ones, speed factors, and weight factors and parameters for the combination functions, are still missing in this matrix, and they are essential too; for example, see Sect. 2.2.3 and the quote from Pearl (2000) there. So, additional information on speed factors and on combination functions and their parameters is needed as well. Moreover, in many cases connection matrices are not very efficient, as usually each state in a network has only a limited number of connections, and then a connection matrix consists mainly of empty cells, which makes the space versus information ratio rather inefficient.

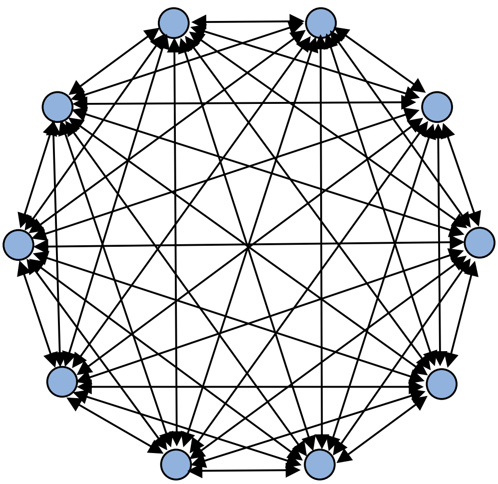

Fig. 2.5 The example Social Network

Table 2.5 Connection matrix of the example Social Network

Connection matrix	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1		0.1	0.2	0.1	0.2	0.15	0.1	0.25	0.25	0.1
X_2	0.25		0.25	0.2	0.1	0.2	0.15	0.25	0.25	0.25
X_3	0.1	0.25		0.1	0.2	0.15	0.1	0.25	0.1	0.15
X_4	0.25	0.15	0.25		0.15	0.8	0.25	0.15	0.25	0.25
X_5	0.25	0.2	0.1	0.2		0.25	0.2	0.1	0.2	0.15
X_6	0.25	0.1	0.25	0.25	0.25		0.1	0.25	0.25	0.1
X_7	0.2	0.1	0.2	0.15	0.2	0.2		0.2	0.15	0.25
X_8	0.1	0.25	0.1	0.25	0.05	0.15	0.25		0.1	0.25
X_9	0.25	0.15	0.25	0.15	0.2	0.1	0.2	0.15		0.15
X_{10}	0.2	0.25	0.2	0.2	0.1	0.2	0.15	0.8	0.2	

2.4.1 Role Matrices as a Specification Format

To get a more complete and uniform, and more compact specification format, as an alternative to connection matrices, *role matrices* are introduced, according to the role played by the specified information. For example, as can be seen in Eq. (2.1) in Sect. 2.3.1, the numbers for the $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\eta}_Y$ and the function $\mathbf{c}_{i,Y}(..)$ and parameters $\boldsymbol{\pi}_{j,i,Y}$ of it play completely different roles. These roles are made more explicit and neatly grouped below by the different role matrices in which they are specified. They cover the main elements of network structure (a) connectivity, (b) aggregation, and (c) timing as indicated in Sect. 2.2.3 above.

Note that the role matrices indeed have a more compact format than connection matrices, and also specify an ordering, which is important as combination functions used to aggregate the impact from multiple connections are not always symmetric in their arguments. Five roles are distinguished and there are five role matrices accordingly (for a first example, see Box 2.1). Here **mb** and **mcw** cover connectivity, **mcfw** and **mcfp** cover aggregation, and **ms** covers timing.

Note that for all role matrices, the first dimension, displayed as the vertical axis, is for the states of the network. In the row of a given state, the other states or values are listed that according to the role specified by that matrix affect this given state:

- **mb** *for the base network connectivity role*

Role matrix **mb** specifies on each row for a given state from which states it has incoming connections. The first (vertical) dimension is for states X_j and the second (horizontal) dimension for the list of states X_i from which the considered state X_j gets incoming connections: the names of these states X_i are indicated in the cells in the row of X_j. This information plays the role of the *base connectivity*. This matrix contains the information in graphical form specified by the arrows in a network picture.

- **mcw** *for the connection weights role*

Role matrix **mcw** specifies on each row for a given state X_j which are the connection weights for the states indicated in the corresponding cells in the base connectivity matrix **mb**. This information plays the role of the *connection weights.*

- **ms** *for the speed factors role*

Role matrix **ms** specifies for each state X_j its speed factor. This matrix has only one column. The first (vertical) dimension is for states and the second (horizontal) dimension for the column with speed values for each state. This information plays the role of the *speed factor.*

- **mcfw** *for the combination function weights role*

Role matrix **mcfw** specifies for each state X_j which basic combination functions $\mathrm{bcf}_i(..)$ are used for it and with which weights $\gamma_{i,Y}$. The first (vertical) dimension is for states X_j and the second (horizontal) dimension for combination functions. This information $\gamma_{i,Y}$ plays the role of the *combination function weights.* A nonzero weight implies that the indicated combination function is used for that state. It is possible that for a given state there are nonzero weights for more than one combination function. This expresses that a weighted sum of multiple combination functions $\mathrm{bcf}_i(\ldots)$ is used as combination function $\mathbf{c}_Y(\ldots)$ for that state:

$$\begin{aligned}&\mathbf{c}_Y(\pi_{1,1,Y}, \pi_{2,1,Y}, \ldots, \pi_{1,m,Y}, \pi_{2,m,Y}, V_1, \ldots, V_k)\\&= \frac{\gamma_{1,Y}\mathrm{bcf}_1\left(\pi_{1,1,Y}, \pi_{2,1,Y}, V_1, \ldots, V_k\right) + \cdots + \gamma_{m,Y}\mathrm{bcf}_m\left(\pi_{1,m,Y}, \pi_{2,m,Y}, V_1, \ldots, V_k\right)}{\gamma_{1,Y} + \cdots + \gamma_{m,Y}}\end{aligned} \tag{2.3}$$

For example, if $m = 2$, $\mathrm{bcf}_1\left(\pi_{1,1,Y}, \pi_{2,1,Y}, V_1, \ldots, V_k\right)$ is the function **eucl(..)** and $\mathrm{bcf}_2\left(\pi_{1,1,Y}, \pi_{2,1,Y}, V_1, \ldots, V_k\right)$ the function **alogistic(..)**, and $\gamma_{1,Y} = 3$, $\gamma_{1,Y} = 1$, then the outcome is

$$\begin{aligned}&\frac{\gamma_{1,Y}\mathrm{bcf}_1\left(\pi_{1,1,Y}, \pi_{2,1,Y}, V_1, \ldots, V_k\right) + \gamma_{2,Y}\mathrm{bcf}_2\left(\pi_{1,m,Y}, \pi_{2,m,Y}, V_1, \ldots, V_k\right)}{\gamma_{1,Y} + \gamma_{2,Y}}\\&= \frac{3\,\mathbf{eucl}\left(\pi_{1,1,Y}, \pi_{2,1,Y}, V_1, \ldots, V_k\right) + \mathbf{alogistic}\left(\pi_{1,m,Y}, \pi_{2,m,Y}, V_1, \ldots, V_k\right)}{4}\\&= 0.75\,\mathbf{eucl}\left(\pi_{1,1,Y}, \pi_{2,1,Y}, V_1, \ldots, V_k\right) + 0.25\,\mathbf{alogistic}\left(\pi_{1,m,Y}, \pi_{2,m,Y}, V_1, \ldots, V_k\right)\end{aligned}$$

- **mcfp** *for the combination function parameters role*

Role matrix **mcfp** specifies for each state X_j and each combination function, the parameters $\pi_{i,j,Y}$ of this combination function for state X_j. Note that this is a 3D matrix with as usual the first (vertical) dimension for the states, the second dimension for the parameters of the combination function, the third dimension for the combination function. This information plays the role of the *combination function parameter values.*

The first two role matrices **mb** and **mcw** can be considered a kind of more compact reformulation of the square matrices for connectivity and connection weights. However, there are two important differences. First, in the connectivity role matrix **mb** the row for a given state X_j displays the names of the states from which X_j gets incoming connections (not the outgoing connections); in a square connection matrix the elements of such a row are in the column for X_j. Second, the connection weights have a separate role matrix **mcw** with numbers for the weights in exactly the cells indicated in the base matrix **mb** for that connection.

Box 2.1 shows the complete specification by role matrices of the conceptual representation of the example network model from Fig. 2.3. Here it can be seen that the role matrices (7 × 2 = 14) are much more compact than connection matrices (7 × 7 = 49), so they are much more efficient as representation. In Box 2.2 this is shown for the example Social Network model of Fig. 2.5. For a fully connected network this condensation is just a modest improvement in efficiency of representation, but usually, networks have many more nodes than the ones connected to a given node.

Box 2.1 Conceptual representation of the example of Fig. 2.3 by role matrices

mb **base connectivity**		1	2
X_1	P		
X_2	Q	X_7	
X_3	R	X_1	X_2
X_4	S	X_3	
X_5	T	X_3	
X_6	U	X_4	X_5
X_7	V	X_5	

mcw **connection weights**		1	2
X_1	P		
X_2	Q	1	
X_3	R	0.8	1
X_4	S	0.9	
X_5	T	1	
X_6	U	1	0.7
X_7	V	1	

ms **speed factors**		1
X_1	P	0
X_2	Q	1
X_3	R	0.4
X_4	S	0.3
X_5	T	0.5
X_6	U	1
X_7	V	0.3

mcfw **combination function weights**		1 **eucl**	2 **alogis-tic**
X_1	P	1	
X_2	Q	1	
X_3	R	1	
X_4	S	1	
X_5	T	1	
X_6	U	1	
X_7	V	1	

mcfp **function**		1 **eucl**		2 **alogistic**	
parameter		1 n	2 λ	1 σ	2 τ
X_1	P	1	1		
X_2	Q	1	1		
X_3	R	1	1.8		
X_4	S	1	0.9		
X_5	T	1	1		
X_6	U	1	1.7		
X_7	V	1	1		

In Box 2.1 it is shown in matrix **mcfw** for the combination function weights that for all states the Euclidean combination function **eucl** is chosen. In matrix **mcfp** for combination function parameters it is shown that order $\boldsymbol{n}$ = 1 is selected (which makes it the scaled sum function), and the scaling factors $\boldsymbol{\lambda}$ are indicated; note that they are chosen here as the sums of the weights of the incoming connections as shown in the rows of matrix **mcw** for the connection weights. Finally, matrix **ms** for speed factors just shows all speed factors that were chosen.

Box 2.2 Conceptual representation of the example Social Network model by role matrices (used in the second and third scenario in Fig. 2.5).

mb base connectivity	1	2	3	4	5	6	7	8	9
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_2	X_1	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_3	X_1	X_2	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_4	X_1	X_2	X_3	X_5	X_6	X_7	X_8	X_9	X_{10}
X_5	X_1	X_2	X_3	X_4	X_6	X_7	X_8	X_9	X_{10}
X_6	X_1	X_2	X_3	X_4	X_5	X_7	X_8	X_9	X_{10}
X_7	X_1	X_2	X_3	X_4	X_5	X_6	X_8	X_9	X_{10}
X_8	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_9	X_{10}
X_9	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_{10}
X_{10}	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9

mcw connection weights	1	2	3	4	5	6	7	8	9
X_1	0.25	0.1	0.25	0.25	0.25	0.2	0.1	0.25	0.2
X_2	0.1	0.25	0.15	0.2	0.1	0.1	0.25	0.15	0.25
X_3	0.2	0.25	0.25	0.1	0.25	0.2	0.1	0.25	0.2
X_4	0.1	0.2	0.1	0.2	0.25	0.15	0.25	0.15	0.2
X_5	0.2	0.1	0.2	0.15	0.25	0.2	0.05	0.2	0.1
X_6	0.15	0.2	0.15	0.8	0.25	0.2	0.15	0.1	0.2
X_7	0.1	0.15	0.1	0.25	0.2	0.1	0.25	0.2	0.15
X_8	0.25	0.25	0.25	0.15	0.1	0.25	0.2	0.15	0.8
X_9	0.25	0.25	0.1	0.25	0.2	0.25	0.15	0.1	0.2
X_{10}	0.1	0.25	0.15	0.25	0.15	0.1	0.25	0.25	0.15

mcfw combination function weights	1 eucl	2 alogistic
X_1	1	
X_2	1	
X_3	1	
X_4	1	
X_5	1	
X_6	1	
X_7	1	
X_8	1	
X_9	1	
X_{10}	1	

mcfp function parameter	1 eucl		2 alogistic	
	1 n	2 λ	1 σ	2 τ
X_1	1	1.85		
X_2	1	1.55		
X_3	1	1.8		
X_4	1	1.6		
X_5	1	1.45		
X_6	1	2.2		
X_7	1	1.5		
X_8	1	2.4		
X_9	1	1.75		
X_{10}	1	1.65		

ms speed factors	1
X_1	0.8
X_2	0.5
X_3	0.8
X_4	0.5
X_5	0.5
X_6	0.5
X_7	0.8
X_8	0.5
X_9	0.5
X_{10}	0.5

In the base connectivity role matrix **mb** shown in Box 2.2 for the fully connected Social Network it is seen that each state has incoming connections from all other states. In the connection weight role matrix **mcw** the connection weights for these connections as indicated in **mb** are shown. For example in the row for state X_2 in the cell (2, 4) for the fourth incoming connection of X_2, the number 0.2 indicates that the state X_5 indicated in the corresponding cell (2, 4) in matrix **mb** has a connection to X_2 with weight 0.2.

In the combination function weight role matrix **mcfw** it is indicated that for all states the Euclidean combination function is chosen, with weight 1. In addition, in the combination function parameter role matrix **mcfp** the two parameters $\boldsymbol{n}$ (the order) and $\boldsymbol{\lambda}$ (the scaling factor) of this combination function are indicated for each

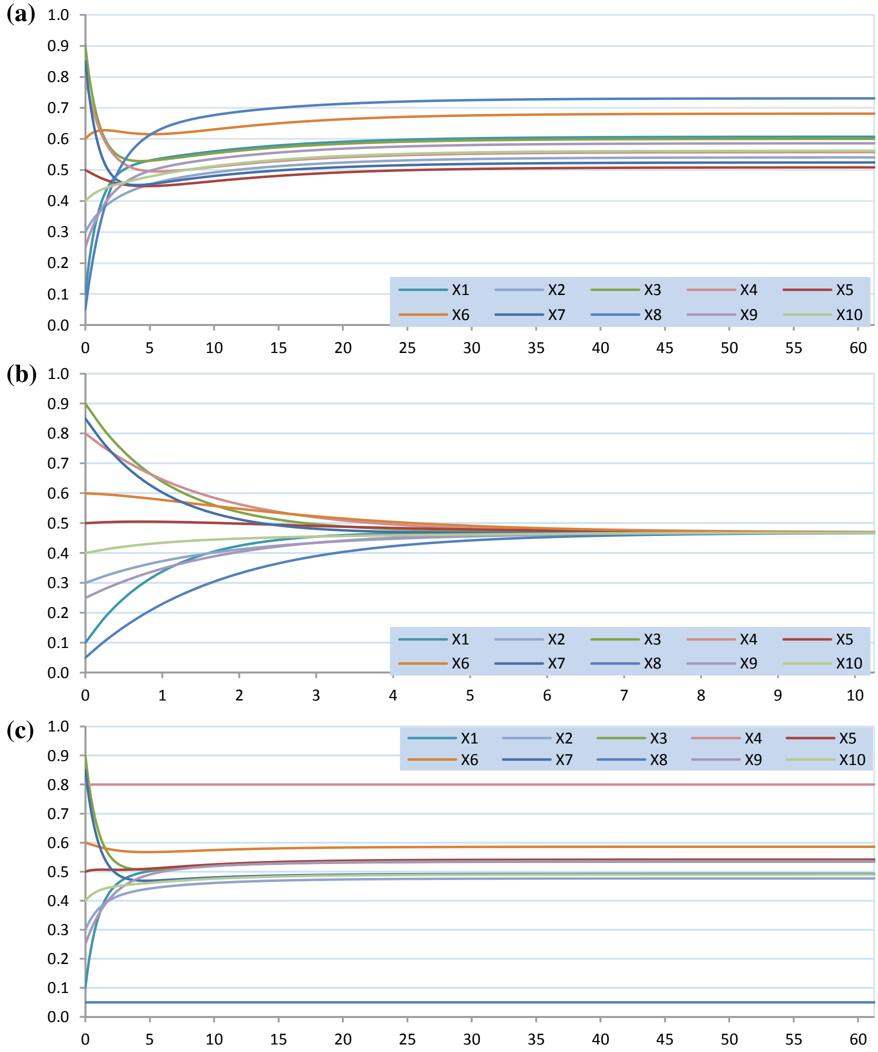

Fig. 2.6 Simulations for the example Social Network of Fig. 2.5 with **a** upper graph: advanced logistic sum combination functions with steepness **σ** = 1.5, threshold **τ** = 0.3, as shown in Table 2.5, right hand side (no convergence to one common value), **b** middle graph: normalised scaled sum functions as shown in Box 2.1, matrix **mcfw** and **mcfp** (convergence to one common value), **c** normalised scaled sum functions with constant X_4 (at 0.8) and X_8 (at 0.05) (no convergence to one common value)

of the states. Finally, the speed factor role matrix **ms** indicates the speed factors of the different states.

In Fig. 2.6 three different simulations are shown for this network model. The first one uses the combination function **alogistic**$_{\sigma,\tau}$**(..)** and the other two the

Table 2.6 Another variant of example role matrices **mcfw** for combination function weights and **mcfp** for combination function parameters, used in the first simulation in Fig. 2.6

mcfw combination function weights	1 eucl	2 alogistic
X_1		1
X_2		1
X_3		1
X_4		1
X_5		1
X_6		1
X_7		1
X_8		1
X_9		1
X_{10}		1

mcfp function	1 eucl		2 alogistic	
parameter	1	2	1 σ	2 τ
X_1			1.5	0.3
X_2			1.5	0.3
X_3			1.5	0.3
X_4			1.5	0.3
X_5			1.5	0.3
X_6			1.5	0.3
X_7			1.5	0.3
X_8			1.5	0.3
X_9			1.5	0.3
X_{10}			1.5	0.3

normalised Euclidean function $\mathbf{eucl}_{1,\lambda}(..)$ of order 1 (which is the normalised scaled sum function). The role matrices for the combination function weights and parameters in Box 2.1 show the second variant. For the first variant, the role matrices in Table 2.6 are used for **mcfw** and **mcfp** instead of those in Box 2.1.

2.4.2 From Network Structure to Network Dynamics: How Role Matrices Define the Basic Difference and Differential Equations

The role matrices contain the values for all of the network structure characteristics that are used to define the basic difference or differential equations describing the network's dynamics based on Eq. (2.1) in Sect. 2.3.1. Actually, there is a direct derivation of the basic difference or differential equations for the different states from the role matrices. This is found as shown in Chap. 10, Sect. 10.6, Box 10.6. It can be seen there, that the equation is indeed fully defined by the role matrices. This derivation shows how the network structure characteristics determine the network's dynamics; see also Fig. 2.7. This is a relation between the network's structure and its dynamics at a basic level. In Chaps. 11–14 this relation between structure and dynamics will be analysed in more detail and at the higher level of properties of structure that entail properties of emerging behaviour (e.g., see Fig. 11.1 in Chap. 11).

Note that, although it may look a bit theoretical, from a practical perspective this is a very relevant derivation. It makes that the design of a network model can fully concentrate on the conceptual representation of the network's structure. As the numerical representation for the network's dynamics fully depends on that, the software environment developed (described in Chap. 9) just takes the role matrices as input and runs the model based on the implied difference equations generated internally by the software, without having to write or even see these

Table 2.7 Specification of initial values

Initial values

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0.1	0.3	0.9	0.8	0.5	0.6	0.85	0.05	0.25	0.4

equations. So, for modeling networks in practice and exploring their behaviour no programming is needed, and even no difference equations need to be specified. Also for mathematical analysis of the network behaviour, usually the difference equations need not to be analysed, as a very simple criterion is available in terms of the network characteristics, that can be used; see also Sect. 2.5 below, and in Chaps. 11–14.

2.4.3 *Simulations for the Example Social Network Model*

For the above social network model, which models social contagion, for example, of opinions or emotions, simulations have been performed for different combination functions. Initial values were used as shown in Table 2.7.

In Fig. 2.7 the three different simulations are shown, all with step size Δt = 0.25. For the upper graph, advanced logistic sum combination functions were used, for the middle graph normalized scaled sum functions, and in the lower graph scaled sum functions while two states remain constant (they have no incoming connections this time, so the cells in the columns for X_4 and X_8 in **mcw** all are 0 now). How can we explain these differences in emerging behavior from the structure of the networks? In Sect. 2.5 these results and their comparison are discussed and analysed in some more detail.

2.5 Relating Emerging Network Behavior to Network Structure

The Network-Oriented Modeling approach based on temporal-causal networks does not only provide opportunities for simulation but also for mathematical analysis and to derive general theoretical results that predict or reflect behavior that is observed in specific cases of simulations. A general question for dynamic models is what patterns of behaviour will emerge, and how their emergence depends on the chosen

Fig. 2.7 Network structure determines network behaviour

structure. Whether or not, in general, such relations between structure and emerging behavior can be found is sometimes a topic for discussion. However, in the context of the Network-Oriented Modeling approach based on temporal-causal networks considered here at least some results on this relation have been obtained.

Usually the structure of a network is described by a number of characteristics. For temporal-causal networks, in particular, such network structure characteristics are connectivity (in terms of connection weights), aggregation (in terms of combination functions) and timing (in terms of speed factors). So, the challenge is to find out how properties of connection weights, combination functions, and speed factors relate to emerging behavior.

2.5.1 Emerging Network Behaviour and Network Structure

Emerging network behaviour can be of different types. Three types are often distinguished:

- **Reaching an equilibrium**

A so called *equilibrium state* is reached, in which for all states the values do not change anymore. This often happens; for example, all three graphs in Fig. 2.6 show examples of this type.

- **Ending up in a limit cycle**

The behaviour ends up in a regular repeating pattern of values (a periodic pattern) for the states; this is called a *limit cycle*. In Fig. 2.8 an example of this is shown, taken from Treur (2016), Chap. 12.

- **Chaotic behaviour**

The behaviour is usually (loosely) called *chaotic* if there is no observed regularity in it. This means that at least no equilibrium is reached and also no periodic pattern as a limit cycle. Lorenz (1963) used as title for his paper on chaotic behaviour 'Deterministic Nonperiodic Flow'. In Mathematics, the area of Chaos Theory has developed more specific definitions for chaotic behaviour, usually involving that the outcome is very sensitive for the values of the initial settings; e.g., Lorenz (1963): the present determines the future but the approximate present does not approximately determine the future. An often cited example or metaphor is that a butterfly at one place in the world can cause a tornado somewhere else (the butterfly effect).

When all state values are in a bounded interval, for example, the [0, 1] interval, most often the first type of emerging behaviour is observed, but sometimes also the other two types can occur. An example (seemingly) showing the third type of emerging behaviour may be found in Chap. 6. Note that a pattern can initially look like this last type, but later on may still turn out to be one of the other two types.

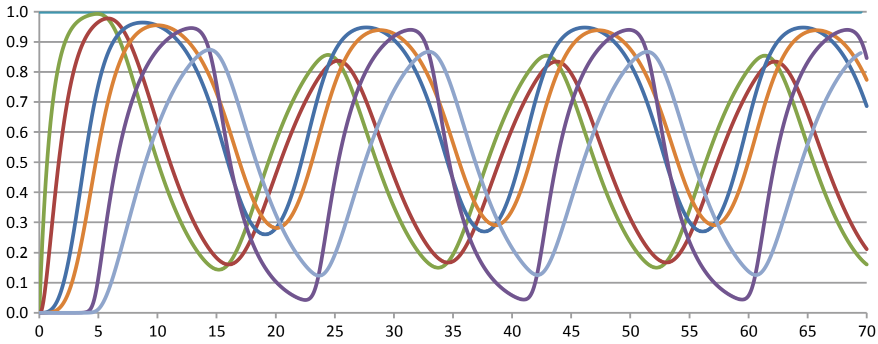

Fig. 2.8 Example simulation ending up in a limit cycle; adopted from Treur (2016), Chap. 12, Sect. 12.7, Fig. 12.7, p. 344

By mathematical analysis of the relation between network structure and network behaviour, the first type of emerging network behaviour (reaching an equilibrium) is relatively easy to explore; see also Treur (2016), Chap. 12. Below some of the basics for that are summarised. In particular, it can be addressed what values eventually will emerge; for example:

- Will the values of each state in the network separately in the end become constant?
- Will the values of different states eventually converge to a common value?
- Under which conditions on the structure of the network will this happen?

Such behaviour relates to what are called stationary points and equilibrium states, defined as follows:

Definition (stationary and equilibrium)

State Y is *stationary* or *has a stationary point* at time t if $\mathbf{d}Y(t)/\mathbf{d}t = 0$.

The network is in an *equilibrium state* at t if all states are stationary at t.

Note that a state may have a stationary point at some time point t, but later on still change its value; in particular, this happens when other states do not have a stationary point at that same time point t. For example, all peaks and dips in Fig. 2.8 indicate stationary points for the specific states, but no equilibrium occurs.

For a temporal-causal network, in particular, there is a simple *criterion* in terms of the network structure characteristics (speed factors $\boldsymbol{\eta}_Y$, connection weights $\boldsymbol{\omega}_{X_i,Y}$, and combination functions $\mathbf{c}_Y(..)$); this immediately follows from Eq. (2.1) in Sect. 2.3.1:

Criterion for stationary point and equilibrium

In a temporal-causal network model, state Y is stationary at t if and only if

$$\boldsymbol{\eta}_Y = 0 \quad \text{or} \quad \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) = Y(t) \tag{2.4}$$

where X_1, …, X_k are the states from which Y has incoming connections.

The network is in *equilibrium* when for *all* states Y of the network this equation holds. These equations are called *equilibrium equations*.

Assuming $\boldsymbol{\eta}_Y > 0$, the equilibrium equations can just be written down from two of the three the network structure characteristics defining the network structure: connection weights $\boldsymbol{\omega}_{X,Y}$ and combination functions $\mathbf{c}_Y(..)$. As an example, in case the combination function is a scaled sum function, such an equation looks like

$$\frac{\boldsymbol{\omega}_{X_1,Y}X_1(t) + \cdots + \boldsymbol{\omega}_{X_k,Y}X_k(t)}{\boldsymbol{\lambda}} = Y(t) \tag{2.5}$$

Often the t is left out as it is a relation between the (constant) values; so the equilibrium equation becomes just an equation for these values:

$$\frac{\boldsymbol{\omega}_{X_1,Y}X_1 + \cdots + \boldsymbol{\omega}_{X_k,Y}X_k}{\boldsymbol{\lambda}} = Y$$

As an example, from the role matrices **mb** and **mcw** for base connectivity and connection weights, and **mcfp** for combination function parameters in Box 2.1 above it can be found that when using the scaled sum combination function in the above example network the equilibrium equation for X_3 is

$$\frac{0.8\,X_1 + X_2}{1.8} = X_3$$

which can be rewritten as

$$0.8\,X_1 + X_2 = 1.8\,X_3 \tag{2.6}$$

So, if in a simulation an equilibrium is reached, then the state values found should satisfy this relation (and also the other equilibrium equations). If not, then something is wrong and has to be resolved.

As another example, from the role matrices **mb** and **mcw** for base connectivity and connection weights, and **mcfp** for combination function parameters in Box 2.2 above it can be found that when using the scaled sum combination function in the above example Social Network the equilibrium equation for X_2 is

$$X_2 = \frac{0.1\,X_1 + 0.25\,X_3 + 0.15\,X_4 + 0.2\,X_5 + 0.1\,X_6 + 0.1\,X_7 + 0.25\,X_8 + 0.15\,X_9 + 0.25\,X_{10}}{1.55} \tag{2.7}$$

or

$$1.55\,X_2 = 0.1\,X_1 + 0.25\,X_3 + 0.15\,X_4 + 0.2\,X_5 + 0.1\,X_6 + 0.1\,X_7 + 0.25\,X_8 + 0.15\,X_9 + 0.25\,X_{10} \tag{2.8}$$

The equilibrium equations are a useful means to *verify the correctness* of the (implemented) network model. This can be done in two ways, as also described in Treur (2016), Chap. 12. The first is by taking the state values as observed for a stationary point or equilibrium in a simulation example, and substitute them in the equilibrium equations. If a serious deviation is found, that should be a reason to investigate the implemented model further to find and resolve some error. Another way, for an equilibrium, is to solve the equilibrium equations and compare the values found with the values observed in a simulation. Whether or not this can be done in an algebraic manner depends on the specific combination functions. These are very practical ways of using the relation of the network structure as specified by the role matrices (based on which the equilibrium equations are formulated) with emergent behaviour as generated by an implemented network model.

Also in a more general sense the relation between network structure and network behavior can be explored. A number of general properties of network structure have been identified such that they relate to similar emergent behavior. These network structure properties concern a connectivity property about how many states of the network are reachable from a given state, and some properties of combination functions. This will be briefly discussed in Sect. 2.5.2.

2.5.2 *Network Structure Properties Relevant for Emerging Network Behaviour*

It has been found out that some properties of network structure (in particular concerning aggregation and connectivity) underly the differences in emerging behaviour shown in Fig. 2.6. Chapters 11 and 12 address this in much more detail. In the current section just a brief introduction and summary is presented. First the relevant properties of aggregation, as specified by combination functions, that have been identified.

Definition (properties of combination functions)

Let c(V_1, …, V_k) be a function of values V_1, …, V_k

(a) c(..) is *nonnegative* if c(V_1, …, V_k) $\geq$ 0
(b) c(..) *respects* 0 if $V_1, \ldots, V_k \geq 0 \Rightarrow [\mathrm{c}(V_1, \ldots, V_k) = 0 \Leftrightarrow V_1 = \cdots = V_k = 0]$
(c) c(..) is *monotonically increasing* if

$$U_i \leq V_i \text{ for all } i \Rightarrow \mathrm{c}(U_1, \ldots, U_k) \leq \mathrm{c}(V_1, \ldots, V_k)$$

(d) c(..) is *strictly monotonically increasing* if

$$U_i \leq V_i \text{ for all } i, \text{and } U_j < V_j \text{ for at least one } j \Rightarrow \mathrm{c}(U_1, \ldots, U_k) < \mathrm{c}(V_1, \ldots, V_k)$$

(e) c(..) is *scalar-free* if $\mathrm{c}(\boldsymbol{\alpha} V_1, \ldots, \boldsymbol{\alpha} V_k) = \boldsymbol{\alpha}\, \mathrm{c}(V_1, \ldots, V_k)$ for all $\boldsymbol{\alpha} > 0$

The properties (a–c) are basic properties expected from most if not all combination functions. Properties (d) and (e) define a specific class of combination functions; this class includes all Euclidean combination functions, but logistic combination functions do not belong to this class as they are not scalar-free. In Chaps. 11 and 12 some theoretical results on emergent behaviour will be presented for this class, where also some other network properties concerning the network's connectivity and normalisation play a role.

Definition (normalised network)
A network is *normalised* or uses normalised combination functions if for each state Y it holds $\mathrm{c}_Y(\boldsymbol{\omega}_{X_1,Y}, \ldots, \boldsymbol{\omega}_{X_k,Y}) = 1$, where $X_1, \ldots, X_k$ are the states with outgoing connections to Y.

Note that $\mathrm{c}_Y(\boldsymbol{\omega}_{X_1,Y}, \ldots, \boldsymbol{\omega}_{X_k,Y})$ is an expression in terms of the parameter(s) of the combination function and $\boldsymbol{\omega}_{X_1,Y}, \ldots, \boldsymbol{\omega}_{X_k,Y}$. To require this expression to be equal to 1 provides a constraint on these parameters: an equation relating the parameter value(s) of the combination functions to the network structure characteristics $\boldsymbol{\omega}_{X_1,Y}, \ldots, \boldsymbol{\omega}_{X_k,Y}$. To satisfy this property, often the parameter(s) can be given suitable values. For example, for a Euclidean combination function, scaling factor $\boldsymbol{\lambda}_Y = \boldsymbol{\omega}^n_{X_1,Y} + \cdots + \boldsymbol{\omega}^n_{X_k,Y}$ will provide a normalised network. This can be done in general:

(1) **normalisation by adjusting the combination functions**

If any combination function $\mathbf{c}_Y(..)$ is replaced by $\mathbf{c'}_Y(..)$ defined as

$$\mathbf{c}'_Y(V_1, \ldots, V_k) = \mathbf{c}_Y(V_1, \ldots, V_k)/\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}, .., \boldsymbol{\omega}_{X_k,Y})$$

then the network becomes normalised: indeed $\mathbf{c}'_A(\boldsymbol{\omega}_{X_1,Y}, .., \boldsymbol{\omega}_{X_k,Y}) = 1$

(2) **normalisation by adjusting the connection weights (for scalar-free combination functions)**

For scalar-free combination functions also normalisation is possible by adapting the connection weights; define:

$$\boldsymbol{\omega}'_{X_i,Y} = \boldsymbol{\omega}_{X_i,Y}/\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}, .., \boldsymbol{\omega}_{X_k,Y})$$

Then the network becomes normalised; indeed it holds:

$$\mathbf{c}_Y(\boldsymbol{\omega}'_{X_1,Y},\ldots,\boldsymbol{\omega}'_{X_k,Y}) = \mathbf{c}(\boldsymbol{\omega}_{X_1,Y}/\mathrm{c}_Y(\boldsymbol{\omega}_{X_1,Y},\ldots,\boldsymbol{\omega}_{X_k,Y}),\ldots,\boldsymbol{\omega}_{X_k,Y}/\mathrm{c}(\boldsymbol{\omega}_{X_1,Y},\ldots,\boldsymbol{\omega}_{X_k,Y})) = 1$$

Another important determinant for emerging behaviour is connectivity: in how far the network has paths connecting any two states; for this, the following definition is used:

Definition State Y is *reachable* from state X if there is a directed path from X to Y

This property makes a difference between the third example simulation and the other two: from no state X_4 or X_8 is reachable in that third case as these states have no incoming connections.

2.5.3 Relating Network Structure Properties to Emerging Network Behaviour

Part of the mathematical analysis performed is summarised by the following theorem that has been derived.

Theorem 1 (equal equilibrium state values)
Suppose a network with nonnegative connections is based on normalised, strictly monotonically increasing and scalar-free combination functions.

(a) Suppose any state Y except at most one state, is reachable from all other states X. Then in an equilibrium state, all states have the same state value.
(b) Under the conditions of (a), the equilibrium state is attracting, and the common equilibrium state value lies in between the highest and lowest previous or initial state values.

Theorem 1 can be used to prove for many cases that in an equilibrium state all states have the same value. This includes cases in which the only combination functions used are Euclidean combination functions. Returning to the example simulations shown in Fig. 2.6, it turns out that in one case convergence to one common equilibrium value takes place, but in the other two cases that does not happen and instead some form of clustering seems to take place. How can we explain these differences in emerging behavior from the structure of the networks? This question can be answered based on the above properties. They show why for the second simulation in Fig. 2.6 convergence to one common value takes place, but not for the first and third case. The first case does not satisfy the scalar-free condition, and the third case does not satisfy the condition on reachability in Theorem 1; one exception is allowed but not two, as occurs in the third example in Fig. 2.6. In case of only one of X_4 and X_8 as exception, say X_4, there would be convergence to one common value: to the value of the one state that remains constant all the time. Note that these differences in emerging behaviour have no relation to linear or nonlinear equations, as Theorem 1 applies to, for example, all Euclidean combination functions, both to linear and nonlinear ones. This type of analysis will be addressed in much more detail and considering more types of functions and network connectivity in Chaps. 11 and 12.

2.6 The Wide Applicability of Network-Oriented Modeling

Many applications of Network-Oriented Modeling exist: Biological Networks, Neural Networks, Mental Networks, and Social Networks. It sometimes is a silent assumption that a Network-Oriented Modeling approach can only work for such specific application domains, where networks are felt as more or less already given or perceived in the real world. It has turned out that the applicability of Network-Oriented Modeling goes far beyond such domains as will be discussed below.

2.6.1 Network-Oriented Modeling Applies Beyond Perceived Networks

In Treur (2017) it is shown that the above-mentioned silent assumption is not a correct assumption. It has been shown that the applicability of the Network-Oriented Modeling approach based on temporal-causal networks is much wider. For example, it has been proven that modeling by temporal-causal networks subsumes modelling approaches based on the dynamical system perspective (Ashby 1960; Port and van Gelder 1995) or systems of first-order differential equations; see Treur (2017), Sect. 2.3. The dynamical system approach is not only often used to obtain dynamical cognitive models, but also to model processes in many other scientific domains, including biological and physical domains. Moreover, modeling by temporal-causal networks subsumes modelling approaches based on discrete (event) and agent simulation (Sarjoughian and Cellier 2001; Uhrmacher and Schattenberg 1998), including very basic computational notions such as finite state machines and transition systems; see Treur (2017), Sect. 2.4.

This shows that temporal-causal network models do not just model networks considered as given in the real world, but can be applied to model practically any type of process. Therefore, indeed the modelling approach is not limited only to Biological Networks, Neural Networks, Mental Networks, and Social Networks, but applies far beyond those types of domains. It shows that the specific temporal-causal interpretation and structure added to networks on top of a basic graph structure, as discussed in Sect. 2.2, does not introduce limitations compared to other dynamic modeling approaches that are based on difference or differential equations.

2.6.2 Network-Oriented Modeling Applies to Network Adaptation

In Chap. 1 it already has been pointed out how adaptive networks can be modelled too, using the notion of network reification. In this book, starting in Chap. 3, in different chapters, this is illustrated for many examples varying from adaptive Mental Networks based on a Hebbian learning principle to adaptive Social Networks based on a homophily principle, and more. Note that this illustrates once more how the presented Network-Oriented Modeling approach provides a unifying perspective across different domains, in this case, the mental domain for Hebbian learning and the social domain for bonding by homophily. Both can be described in a unified manner by a picture of the type as shown in Fig. 2.3, and by some combination function specifying the specific adaptation principle: see Chap. 3, Sect. 3.6.1 and Fig. 3.4.

2.7 Discussion

In this chapter, the ins and outs of the Network-Oriented Modeling perspective were discussed in some detail. Part of the material for this chapter is based on Treur (2019).

By committing to an interpretation of networks based on the notion of a temporal-causal network, more structure and more depth is obtained, and more dedicated support is possible. At first sight, it may suggest that it introduces a limitation to commit to a specific interpretation and structure of networks, but the proven wide scope of applicability of this Network-Oriented Modelling approach shows otherwise, as causality and temporality are very general concepts; e.g., see also Treur (2016, 2017). On the contrary, the specific network structure characteristics connection weights, combination functions, and speed factors allow for a quite sensitive and unifying way of modeling realistic processes. These characteristics also allow more theoretical depth, which was illustrated by presenting some mathematical results on how emerging network behaviour relates to specific properties of the network structure.

In the rest of the book, a number of fundamental themes are being developed in more depth. A major theme is *network reification* as already pointed out in Chap. 1. This provides a substantial enhancement of expressive power of the modelling format, in particular where it concerns adaptive networks where the network structure characteristics change over time. By considering dynamics not only for states but also for characteristics of the network structure such as connection weights, also adaptive processes are covered in the form of reified adaptive networks. Examples of this illustrate the unifying role that such reified temporal-causal network models can play, in particular by revealing similar structures in adaptive Mental Networks based on a Hebbian learning principle (Hebb 1949; Gerstner and

Kistler 2002), and adaptive Social Networks based on a homophily principle (McPherson et al. 2001). The structure provided by the notion of temporal-causal network in conjunction with the notion of network reification introduced in more depth in Chap. 3 provides the machinery to express such adaptive processes in a unified manner. More specifically, in Chap. 3 it is shown how the network reification construction can be defined in general, and it is illustrated by several examples for Mental and Social Networks how any network adaptation principle can be defined within the reified network. In Chap. 4 it is shown how this reification construction can be repeated, thus obtaining *multilevel network reification* in which, for example, adaptive adaptation principles can be represented explicitly.

Another fundamental theme being developed further in more depth is the *relation between network structure and emerging network behaviour*. Keeping in mind that network structure is defined by network characteristics Connectivity, Aggregation and Timing in terms of connection weights, speed factors, and combination functions, in this theme it is analysed how certain properties of these network structure characteristics relate to certain emerging behaviour (mainly focusing on equilibria). For example, can the values of the states for $t \to \infty$ be predicted from these characteristics? And in which cases will all states end up with the same common value? This is addressed in Chaps. 11 and 12, where in the latter chapter the network is analysed based on its *strongly connected components* (Harary et al. 1965) and *stratification* (Chen 2009) of the abstracted acyclic *condensation graph*. For reified networks for bonding based on homophily these questions are addressed in Chap. 13, and for Hebbian learning, this is addressed in Chap. 14.

A third area in which much development takes place is in the area of applications to certain biological, mental and social domains. The temporal-causal format makes it easy to represent causal domain knowledge in an understandable and executable manner. Several examples of applications in these domains illustrate this. In addition, it may be interesting to further investigate applications to the area of business economics, organisation modeling and management; e.g., Naudé et al. (2008).

References

Ashby, W.R.: Design for a Brain, 2nd edn. Wiley, New York (1960)

Bell, A.: Levels and loops: the future of artificial intelligence and neuroscience. Phil. Trans. R. Soc. Lond. B **354**, 2013–2020 (1999)

Chen, Y.: General spanning trees and reachability query evaluation. In: Desai, B.C. (ed.), Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering, C3S2E'09, pp. 243–252. ACM Press (2009)

Gerstner, W., Kistler, W.M.: Mathematical formulations of Hebbian learning. Biol. Cybern. **87**, 404–415 (2002)

Harary, F., Norman, R.Z., Cartwright, D.: Structural Models: An Introduction to the Theory of Directed Graphs. Wiley, New York (1965)

Hebb, D.: The Organisation of Behavior. Wiley, New York (1949)

Jonker, C.M., Snoep, J.L., Treur, J., Westerhoff, H.V., Wijngaards, W.C.A.: Putting intentions into cell biochemistry: an artificial intelligence perspective. J. Theor. Biol. **214**(2002), 105–134 (2002)

Jonker, C.M., Snoep, J.L., Treur, J., Westerhoff, H.V., Wijngaards, W.C.A.: BDI-modelling of complex intracellular dynamics. J. Theor. Biol. **251**, 1–23 (2008)

Kim, J.: Philosophy of Mind. Westview Press (1996)

Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**(2), 130–141 (1963)

McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. Annu. Rev. Sociol. **27**, 415–444 (2001)

Mooij, J.M., Janzing, D., Schölkopf, B.: From differential equations to structural causal models: the deterministic case. In: Nicholson, A., Smyth, P. (eds.), Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI-13), pp. 440–448. AUAI Press (2013). URL: http://auai.org/uai2013/prints/papers/24.pdf

Naudé, A., Le Maitre, D., de Jong, T., Mans, G.F.G., Hugo, W.: Modelling of spatially complex human-ecosystem, rural-urban and rich-poor interactions (2008). URL: https://www.researchgate.net/profile/Tom_De_jong/publication/30511313_Modelling_of_spatially_complex_human-ecosystem_rural-urban_and_rich-poor_interactions/links/02e7e534d3e9a47836000000.pdf

Pearl, J.: Causality. Cambridge University Press, Cambridge (2000)

Port, R.F., van Gelder, T.: Mind as motion: explorations in the dynamics of cognition. MIT Press, Cambridge, MA (1995)

Potter, S.M.: What can artificial intelligence get from neuroscience? In: Lungarella, M., Bongard, J., Pfeifer, R. (eds.) Artificial Intelligence Festschrift: The next 50 years. Springer, Berlin (2007)

Sarjoughian, H., Cellier, F.E. (eds.): Discrete Event Modeling and Simulation Technologies: A Tapestry of Systems and AI-Based Theories and Methodologies. Springer, Berlin (2001)

Scherer, K.R.: Emotions are emergent processes: they require a dynamic computational architecture. Phil. Trans. R. Soc. B **364**, 3459–3474 (2009)

Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Springer Publishers, Berlin (2016) Downloadable at URL: https://link-springer-com.vu-nl.idm.oclc.org/book/10.1007/978-3-319-45213-5

Treur, J.: On the applicability of network-oriented modeling based on temporal-causal networks: why network models do not just model networks. J. Inf. Telecommun. **1**, 23–40 (2017)

Treur, J.: The ins and outs of network-oriented modeling: from biological networks and mental networks to social networks and beyond. In: Transactions on Computational Collective Intelligence vol. 32, pp. 120–139. Springer Publishers, Berlin. Contents of Keynote Lecture at ICCCI'18 (2019)

Uhrmacher, A., Schattenberg, B.: Agents in discrete event simulation. In: Proceedings of the European Symposium on Simulation (ESS '98, Nottingham, England). Society for Computer Simulation, San Diego, CA (1998)

Westerhoff, H.V., He, F., Murabito, E., Crémazy, F., Barberis, M.: Understanding principles of the dynamic biochemical networks of life through systems biology. In: Kriete, A., Eils, R. (eds.) Computational Systems Biology, 2nd edn, pp. 21–44. Academic Press, Oxford (2014a)

Westerhoff, H.V., Brooks, A.N., Simeonidis, E., García-Contreras, R., He, F., Boogerd, F.C., Jackson, V.J., Goncharuk, V., Kolodkin, A.: Macromolecular networks and intelligence in microorganisms. Front. Microbiol. **5**, Article 379 (2014b)

Wright, S.: Correlation and causation. J. Agric. Res. **20**, 557–585 (1921)